

## METHOD AND APPARATUS FOR PRODUCING AN IMPLANT

This application claims the benefit of U.S. Provisional Application Nos. 60/148,275, 60/148,277, and 60/148,393, which were all filed August 11, 1999.

### Background of the Invention

5           The present invention relates to fabricating implants to replace bony structures. More specifically, the invention relates to a system and methodology for fabricating a "drop in" replacement for a particular segment of missing bony structure, in which the implant fits precisely within the contours of the missing segment and thus minimizes complications during the surgical procedure to install  
10   the implant, and subsequently during the healing process. It will be appreciated, however, that the invention is also amenable to other like applications.

          Various systems and methods of fabricating prosthetic implants are known in the prior art. Examples of such prior systems and methods include U.S. Patent Nos. 4,436,684; 5,274,565; 5,357,429; 5,554,190; 5,741,215; and  
15   5,768,134. Each of these patents, however, suffer from many disadvantages that have collectively limited the usefulness of their methods and implants to the relevant field.

### Neural Network Segmentation

          Although clinical visualization of three dimensional organ surfaces  
20   embedded in an imaged volume is now common, segmentation techniques are often ineffective and time consuming. Current three dimensional (3D) surface segmentation is often dependent on the contrast and resolution of two dimensional (2D) anatomical boundaries identified by an operator proceeding slice by slice

through the image volume. Existing automated 3D surface and volume segmentation algorithms, such as region growing, are most commonly based on voxel intensity, nearest neighbor similarity, and edge strength. An active contours energy minimization approach attempts to automate 2D outline detection. An extension to 3D suggests regions to be segmented by internally enveloping them with three dimensional bubbles. Another method based explicitly on high level volumetric shape features utilizes prior information obtained from previously imaged anatomical objects. This approach is based on prior morphological knowledge and a highly integrated set of model based grouping operations. The use of prior information allows the last method to be more adaptive to an expected surface when it encounters noise due to artifact or non-descript structures such as in thin areas. However, detection of anatomical surfaces as complex as the skull or soft tissue face with prior information alone is computationally expensive. It is likely manual work will be necessary where the signal is too low, especially in regions with high artifact noise or where artifact-interrupted surface morphology may obfuscate a match with database information. The relationship of artifacts to surface morphology is dependent on both surface proximity to the artifact-generating structure and patient pose. To overcome these limitations, a third approach, combines semi-automatic image segmentation procedures with intelligent scissors. A user steers the image segmentation via live wire and live lane tools through the image volume.

Current slice-based segmentation methods tend to be time consuming. Imprecision correlates with the numbers of decisions the operator is forced to make. These are concentrated in areas of poor contrast, artifact, or insufficient geometry for the user to recognize local aspects of the patient's anatomy (i.e., the sufficiency of anatomical cues are heavily dependent on structural size and degree of curvature as well as how patient pose affects the representation in tomographic images). Difficulty in making segmentation decisions based on these three (3) types of information may be compounded by a user environment where an image volume is presented as a series of tomographic or projection images that lose resolution when reformatted. In all cases these two

dimensional views require anticipation of how the segmented outlines will combine to form the 3D surface that the user wishes to render.

Deformable curve fitting for edge detection seems less adaptable than manual segmentation for determining the location of anatomical edges. Curve fitting runs into trouble when edges vary in signal strength, requiring multiple initialization and guidance steps to obtain best-fit. An operator familiar with the salient anatomy quickly adapts to provide continuity between areas of material inhomogeneities (i.e., varying pixel intensity) that may signal a break to an edge detection algorithm. It is possible that these edges can be spanned by the use of geometrically ideal shapes or those based on prior information. However, these simple models are less adaptable, or less decisive, than manual segmentation.

Neural networks, applied to volume image segmentation, are capable of modeling structural components in the image through the processing of the gray scale and spatial information recorded in volume images. They take advantage of the multidimensional nature of medical images. Neural network classification can be either supervised or unsupervised. To date, supervised neural networks have been preferred, despite their dependence on user interaction to train or correct their results. Unsupervised neural networks do not rely on pre-labeled samples. Rather, they require careful selection of input events that, when clustered, are useful to the user.

#### Fitting a Deformable Template to a Segmented Surface

There is debate as to whether comparison of three dimensional (3D) anatomical shapes should be done on the basis of whole objects or elementally. One conventional method for 3D anatomical surface parsing begins with identification of a specialized class of space curves, which are referred to as "ridge curves."

A consensus ridge curve definition would cite a maximum first principal curvature along sharp edges as allowing their representation as a space curve. The curvature found along ridge curves is regular enough that a class of curvature extremal landmarks can be readily detected. These are referred to in the literature as Type II landmarks, which may be extracted manually or automatically.

These landmarks may form the initial seed points on which to register a wire frame (i.e., ridge curve and "geodesic" lines) plate. "Geodesic" space curves, a shortest path traversed on a graphical manifold, are useful as a means to link Type II landmarks. The resulting fitted ridge curve and "geodesic" wireframe are used as a means to tile out and parameterize 3D surface images. This toolkit allows parameterized (labeled) surface extractions, multi-surface registration, and averaging for cephalometric comparisons of the skull, soft tissue face, and cerebral ventricles.

In regards to parameterized 3D surface extraction, the toolkit facilitates an initially manual superimposition of a ridge curve-based deformable template to the perspectively rendered surface of interest at the Type II landmarks. However, there is only an approximate tie between the surface tile points bounded by ridge and "geodesic" space curves, and the surface voxels originally identified (segmented) in the raw CT or MR scan volume image data. The "geodesic" criteria are less appropriate and these lines have been renamed "tiling curves." Additionally, the entire surface is encoded as set of B-spline space curves, in order to facilitate whole surface averaging algorithms, (i.e., to average multiple surfaces space curve by space curve). The toolkit links biological labeling (homology mapping) and geometric surface subdivision, resulting in a homology mapped (tile name) parameterized ( $u, v$  space of tile) surface.

The primary basis for positioning multiple biological surfaces for conventional averaging is a specialized class of space curves referred to as "ridge curves". One definition of ridge curves describes them as biologically significant space curves found along sharp edges, corresponding to first principal curvature maxima. These sharp edges are the most bent part of the surface. Type II landmarks, identified as recurrent curvature maxima (bends), are found along ridge curve space curves. They span these landmarks with what are called "geodesics" in order to create three or four sided tiles ("tiling curves").

It has been proposed to add tile parameters (i.e., tile  $u, v$  coordinates and tile name) to the xyz point coordinates found in 3D skull surface images to produce a feature-preserving average or morphometric comparison of homologous

landmark. A geometric subdivision (ridge curve and geodesic tile boundaries) of the surface links biological labels (homology mapping) to tile grid ( $u$ ,  $v$ ) parameterization of the extracted points ( $x$ ,  $y$ ,  $z$ ). The toolkit begins superimposition of the ridge curve and geodesic (tiling curve) deformable wireframe template with the identification of Type II landmarks. Using a thin plate spline, the ridge curve and geodesic template wireframe is warped to the surface's Type II landmarks. Superimposition of the interceding ridge curve and geodesic wire segments to best position proceeds automatically.

### Averaging of Multiple Tiled Out Specimens

When too much of the patient is missing from the image, it is useful to match the average to the patient to obtain a reasonable shape to fill the defect. Unlike images of individuals, averages are usually relatively smooth and symmetric (right to left). However, estimating the shape of a defect of large size with an average of too few subjects is also not helpful. In this case, it may be better to use the image of a single relative.

Homology-based 3D surface averaging requires appropriate superimposition of sample member surface points within a shared reference frame. The toolkit obtains this position in a series of global warping and local unwarping maneuvers. First an average of all sample specimen's Type II landmarks is obtained. Each specimen is then globally warped to these landmarks and then locally unwrapped. A rough average of the ridge and geodesic curve sample is produced. Next, because the space curve sample members are fit to the resulting rough average space curve, perpendicular bisectors may be positioned equidistantly along the rough average. An average of the intercepted points appropriately samples the shape of each space curve on the entire surface. The  $u, v$  parameters determine the number of points along either side of the four (4) sided tiles and, therefore, contain  $u \times v$  internal space curves, which are averaged in the same manner as the boundary curves. This method insures that these sampling planes (perpendicular bisectors) are never parallel to the sampled data.

30           The present invention provides a new and improved apparatus and method which overcomes the above-referenced problems and others.

### **Summary of the Invention**

A method for determining a shape of a medical device to be implanted into a subject produces an image including a defective portion and a non-defective portion of a surface of a tissue of interest included in the subject.

5 The tissue of interest is segmented within the image. A template, representing a normative shape of an external anatomical surface of the tissue of interest, is superimposed to span the defective portion. An external shape of an implant, is determined as a function of respective shapes of the defective portion as seen in the template, for repairing the defective portion.

10 In accordance with one aspect of the invention, a volumetric image of the tissue of interest is produced.

In accordance with a more limited aspect of the invention, one of a CT and MR image is produced.

In accordance with another aspect of the invention, a position for seating the implant into the defective portion is determined. The implant is tapered as a function of a seating strategy and a space available for the implant.

In accordance with another aspect of the invention, the template is superimposed to the surface of the tissue of interest via warping.

20 In accordance with another aspect of the invention, the template is warped to an external surface of the non-defective portion of the tissue of interest.

In accordance with another aspect of the invention, a ridge and tiling curve homology map is produced of another anatomical surface. The average normative shape is produced from control surface image representations of the anatomical surface of interest.

25 In accordance with a more limited aspect of the invention, the normative shape is determined from a mirror image of the tissue of interest.

In accordance with another aspect of the invention, the normative shape is modified as a function of a shape of the non-defective portion of the anatomical surface of interest.

A system determines a shape of a medical device to be implanted into a subject. An imaging device produces an image including a defective portion and a non-defective portion of a tissue of interest included in the subject. A segmenting means segments the tissue of interest. A template spans the defective portion. The template represents a normative shape of an external anatomical surface of the tissue of interest. A determining means determines an external shape of an implant, as a function of respective shapes of the defective portion and the template, for repairing the defective portion.

In accordance with one aspect of the invention, the image is a volumetric image of the anatomical surface of interest.

In accordance with a more limited aspect of the invention, the volumetric image is one of a CT and MR image.

In accordance with another aspect of the invention, the template specifies a seated edge, which is fixed to the subject, and a surface shape, which is not affixed to the subject.

In accordance with another aspect of the invention, the template is warped to a deformed bone.

In accordance with another aspect of the invention, the template is warped to an external surface of the non-defective portion of the tissue of interest.

In accordance with another aspect of the invention, the normative shape of the template is determined from an additional tissue representative of the hard tissue of interest.

In accordance with another aspect of the invention, the normative shape of the template is determined as a function of an average shape of the tissue of interest.

A method for repairing a defect in a tissue of interest included in a subject produces a volumetric image showing a defective portion, which includes the defect, and a non-defective portion of the hard tissue of interest within the subject. The tissue of interest is segmented from the image. A template, having an average shape of the tissue of interest, is warped over the defective and non-defective portions. A shape of the implant is determined, as function of respective

shapes of the defective portion and the template. The implant is inserted into the defective portion for repairing the defect.

In accordance with one aspect of the invention, the implant is cast.

In accordance with another aspect of the invention, a mirror of  
5 another tissue of interest, representing a bilateral mirror image of the tissue of interest, is homology mapped. The normative shape is determined from the mirrored other section of tissue.

One advantage of the present invention is that it presents an  
unsupervised neural network algorithm (i.e., self organizing feature map, or  
10 SOFM) for automated and accurate 3D surface segmentation, the latter being a necessity for computer generation of cranial implants, diagnostic cephalometrics and image guided surgery.

Another advantage of the present invention is that user interaction in  
the SOFM algorithm is limited to determining which segmented surfaces represent  
15 the anatomy desired by the user.

Another advantage of the present invention is that it rapidly  
assembles automatically segmented, and anatomically valid (i.e., no correction  
required), surfaces into a surface of interest for rendering of 3D surfaces found in  
volume images.

20 Another advantage of the present invention is that it saves time over manually identifying each pixel or currently available semi-automatic methods.

Another advantage of the present invention is that it locates  
continuous surfaces, even in areas of thin structure or imaging artifact (e.g.,  
metallic artifact in x-ray or CT).

25 Another advantage of the present invention is that it combines spatial and intensity information to track 3D surfaces for segmentation.

Another advantage of the present invention is that it provides a  
means for anatomically valid segmentation (because 3D surfaces are tracked  
oblique to the scan plane, irrespective of patient pose (orientation of scan planes  
30 are determined relative the patient anatomy) in the scanner).





Another advantage of the present invention is that it uses average 3D surface images to model surfaces in patient images for rapid prototyping of prosthetic implants.

5 Still further advantages of the present invention will become apparent to those of ordinary skill in the art upon reading and understanding the following detailed description of the preferred embodiments.

### **Brief Description of the Drawings**

The invention may take form in various components and arrangements of components, and in various steps and arrangements of steps. The  
10 drawings are only for purposes of illustrating a preferred embodiment and are not to be construed as limiting the invention.

FIGURE 1A illustrates a bone of interest according to the present invention;

FIGURE 2 illustrates a flowchart according to the present invention;

15 FIGURE 3 illustrates a local negative feedback which excites neuron drop off allowing coordinated input of simulated neuronal field to a neural network.;

FIGURE 4 illustrates event inputs to a Network;

20 FIGURE 5 illustrates a current implementation of a SOFM neural network configuration shown as flow of event space image element input to processor layer of neural network.;

FIGURE 6 illustrates Gaussian and Zero crossing curves;

FIGURE 7 illustrates processor network configurations according to the present invention;

25 FIGURES 8 and 9 illustrate flowcharts of the segmentation process;

FIGURE 10 illustrates multiple cluster back-projection from map nodes;

FIGURE 11 illustrates a SOFM-based image segmentation and surface generation interface;

FIGURES 12A-12E illustrate a feature cluster map, back projected pixels and triangulated surface rendering; FIGURES 12A and 12B display the resulting triangulated surface rendered soft tissue face and skull. FIGURES 12C and 12D present the voxel back projection from the selected feature cluster nodes;  
 5 FIGURE 12E presents the feature cluster map;

FIGURES 13A and 13B illustrates a self organizing feature map formation process in which an initial random processor location evolves into the intended grid-like structure as ordering is obtained by the neural network;

FIGURE 14 illustrates both sessions of manual and SOFM segmented surfaces for skull surfaces;  
 10

FIGURE 15 illustrates both sessions of manual and SOFM segmented surfaces for soft tissue face surfaces;

FIGURES 16A and 16B illustrate plots of accumulated differences across all five data sets per slice the soft tissue face surface and the skull surface,  
 15 respectively;

FIGURE 17 illustrates orthogonal projections of 2D images;

FIGURE 18 illustrates a new ridge curve-based deformable template wireframes superimposed on a soft tissue face graphical manifold, with Type II landmarks, ridge curve arcs, and tiling curves indicated;

FIGURE 19A and 19B illustrate Type II landmarks of the soft tissue face and boney skull templates;  
 20

FIGURES 20A and 20B illustrate a CT slice highlighting segmented surface voxels of interest which are indexed to vertices (points) in an adjacent rendered manifold surface;

FIGURES 21A and 21B illustrate a candidate wireframe prior to the warp at lower left and the result of the thin plate spline warp of the wireframe onto the Type II landmarks, respectively;  
 25

FIGURES 22A illustrates a space curve  $C$  from a sweeping plane positioned at each control point, with tangent vector  $\vec{t}$ , along a normal section of segmented voxel surface  $S$ ;  
 30

FIGURES 22B illustrates a surface normal curvature  $k_n$  in direction  $\bar{t}$ , where  $k_n$  attains principal curvature values  $k_1$  and  $k_2$ ;

FIGURES 23A and 23B illustrate a plane constructed from the two endpoint landmarks and an object's center of mass;

5           FIGURES 24 illustrates a trace along the candidate template's control point to the image surface;

FIGURE 25A illustrates a cross section on the top of the skull and tiling curve control point search planes intersecting that image surface;

FIGURE 25B illustrates a series of search plane iterations;

10           FIGURE 26A illustrates rotating search planes;

FIGURE 26B illustrates search planes for one ridge curve arrayed along a warped ridge curve-based deformable template space curve;

FIGURE 26C illustrates a final ridge curve registration found by the simulated annealing algorithm;

15           FIGURE 26D illustrates a candidate tiling curve and its associated search planes;

FIGURE 26E illustrates the fitted control points;

FIGURE 27 illustrates an entire final ridge and tiling curve wireframe superimposed on the image surface;

20           FIGURE 28 illustrates a flowchart for minimizing the cost function;

FIGURE 29 illustrates a temperature cooling schedule from an initial temperature 100 to a final temperature, -10;

FIGURES 30A, 30B, and 30C illustrates an overview of a ridge curve-based deformable template registration process in SASE;

25           FIGURE 31 illustrates three orthogonal projection views showing direct differences between two surface extractions;

FIGURES 32A and 32B illustrate the soft tissue face surfaces and boney skulls, respectively, obtained via different methods versus the toolkit on the right side;

FIGURES 33A and 33B illustrate color coded difference images of the skull and soft tissue face surface extraction comparisons with the segmented voxel surface and color coded difference images of SASE skull and soft tissue face surface extractions, respectively;

5                   FIGURES 34A and 34B illustrate a tile point density;  
                     FIGURE 35 illustrates the toolkit and SASE surface extractions;  
                     FIGURE 36A illustrates an alignment of u and v internal tile curves in a 4 sided tile;

                    FIGURE 36B illustrates an alignment of u and v internal tile curves  
 10    in a 3 sided tile;

                    FIGURE 37A illustrates a soft tissue face ridge curve-based deformable template definition;

                    FIGURE 37B illustrates a boney skull deformable template definition;

15                   FIGURE 38A illustrates a translation of two shapes into a shared two dimensional common coordinate system;

                    FIGURE 38B illustrates scaling adjustments between the two shapes;

                    FIGURE 38C illustrates the rotation, in order to obtain the  
 20    Procrustes distance (square root of sum of squared) between homologous point landmarks pairs: A...D, a...d;

                    FIGURE 39 illustrates a space curve averaging procedure;

                    FIGURE 40A illustrates a soft tissue face average surface generation;

25                   FIGURE 40B illustrates a skull average surface generation;

                    FIGURE 41A illustrates a soft tissue face surface average comparison of various methods;

                    FIGURE 41B illustrates a boney skull surface average comparison of various methods;

30                   FIGURE 42A illustrates a surface average tile-by-tile color coded error maps for skull and soft tissue face surface extractions;

FIGURE 42B illustrates an SSA surface average tile-by-tile color coded error maps for skull and soft tissue face surface extraction;

FIGURE 43A illustrates a tile point density for soft tissue face average;

5                   FIGURE 43B illustrates an SSA method tile point density;

FIGURE 43C illustrates a boney skull surface average;

FIGURE 43D illustrates an SSA method skull tile point density; and

FIGURE 44 illustrates an inter-method average surface continuity.

### Detailed Description of the Preferred Embodiments

10                   With reference to FIGURES 1, 1A, and 2, a subject 10, or patient, is imaged using an imaging device 12 in a step A. Preferably, the image is a volumetric image obtained using a computerized tomography ("CT") scanner according to a preferred protocol. However, other images, obtained using other imaging techniques (e.g., magnetic resonance) are also contemplated. Once the  
15                   data is captured from the CT scanner 12, it is stored in a processor 14 (preferably, a transportable storage medium) in a step B (although it could be transferred over a network). Then, the processor 14 segments the data in a step C for extracting a region of the image including a target tissue (e.g., liver, lung, or hard tissue such as bone) 16 of interest within the subject; the extracted region 16 includes two (2)  
20                   portions (i.e., one portion having a defect 18 and one portion 20 without the defect). Preferably, the non-defective portion 20 substantially surrounds the defective portion. An external surface of the extracted region 16 of the image is mapped in a step D. A decision is made in a step E whether to register the mapped data of the non-defective portion of the extracted region to a memory device in the  
25                   processor 14, which includes normative (average) data for various bones as a function of various factors (e.g., race, sex, age, etc) of the subject. If the data is transmitted to the memory device, the data representing points on the mapped surface of the non-defective portion of the extracted region are incorporated into averaged data for the respective bone based on the various factors of the subject in  
30                   a step F.

After the image is segmented and mapped, a template 24 is superimposed in a step G, to an external surface of a normative shape of the target tissue 16 (i.e., without the defect 18). Preferably the template is superimposed via warping. However, a best fit process is also contemplated. The template specifies a "seated" edge, which is fixed to the subject (patient), and the remaining surface shape, which is not affixed to the subject.

It is to be understood that the normative shape represents a desired or average shape of the target tissue. Preferably, the normative shape is determined in a bilateral structure from a substantially mirror image of the target tissue (e.g., a defect is present in a left side of a skull and complete data is available for a right (control) side of a skull). However, if too much of the target tissue is missing on the affected side, or the defect crosses the midline in a way that prevents establishing the midline, appropriate normative average surfaces (which, as described below, are already homology mapped) are used. In this case, the normative average data is brought in and registered to the homology mapped normative surface surrounding the defect. Once the template is warped, points on the template are mapped, in a step H, to points on the external surface of the normative shape of the bone of interest 16.

A shape of an implant 26 to be "dropped in" the defect 18 is determined, in a step I, as a function of a difference between the mapped points on the external surface of the target tissue and the external surface of the template. The implant 26 is seated (fit) in the defect 18 of the target tissue 16 in a step J. A fit of the implant (seating and shape relative to surrounding anatomy) is analyzed in order to produce an image description of the best shape for the implant on computer, not manually. That image description is saved in industry standard format, such as stereolithography (STL), and printed on a 3D rendering device (e.g., stereolithographic printer). The fit is then accomplished using a tapering scheme based on the seating strategy and space available for the implant.

If the implant is not produced in biocompatible material, it is cast in an implantable material. It is to be understood that the present methods work with

inert casting materials (e.g., bioplastics such as Polymethymethacrylate) or future tissue engineered materials.

A stereolithography manufacturing method is contemplated for forming the either the implant master shape or the actual implant.

5

### I. Segmentation

As described below, the step C of segmenting the image is preferably executed using self organizing feature maps ("SOFM"), which are used for computationally encoding neural maps.

The SOFM incorporates a modified Hebbian rule. The resulting mapping of data ordering resembles observed neurological patterns. Since then, the SOFM has found widespread use in many practical engineering applications. The SOFM derives primarily from observations of two biologically equivalent computational hypotheses defined as:

(1) The Hebbian Rule: "A synapse is strengthened or stabilized when pre- and post synaptic activations are correlated." Change in synaptic strength is defined as:

$$s_{t+1} = s_t + \alpha \cdot \eta \quad (1)$$

where  $s_t$  is a connection strength at time  $t$ ,  $\eta$  is an output (IP in Figure 4) equal to  $s_t \times \xi$ ,  $\xi$  is an input pattern, and  $\alpha$  is a factor, a user-selected value, that is made to decrease for each iteration, to ensure it reaches zero. This, in turn, ensures appropriate synaptic strength formation and system stabilization.

(2) Local Feedback: Each neuron laterally influences the neighboring neurons to either an excitatory or inhibitory state. This results in a competition which enables the formation of a map. Figure 3 is an example of an excitation function. The SOFM based 3D surface segmentation presented here extends the concepts of ordered feature maps. In the neural network, each normalized input feature vector is scaled or weighted by multiplication with a user-assigned constant vector. This process begins with an event generation step, which results in a pattern  $\xi = \{\xi_1, \xi_2, \dots, \xi_n\}$  that is sent to a "resonator" from which potential connection are made. The weights determine feature connection

30



strengths. The scaling results in a vector  $\mathbf{x}_i = \mathbf{c} * \mathbf{F}$ , with  $\mathbf{c}$  the feature connection strengths and  $\mathbf{F}$  the input vector are shown in Figure 4. Processor weights adapt to input stimulation and in so doing the neural network organizes an ordered map.

The unit  $i$  has connection weights,  $\mu_{i1}, \mu_{i2}, \dots, \mu_{in}$ , which are expressed as a column vector  $\mathbf{m}_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T$ . The discriminant function of a unit is defined as:

$$\eta_i = \sum_{j=1}^n \mu_{ij} \xi_j = \mathbf{m}_i^T \boldsymbol{\xi} \quad (2)$$

For unit  $k$  after selecting a winner from each iteration, the weight vector is updated as:

$$\mathbf{m}_i(t+1) = \frac{\mathbf{m}_i(t) + \alpha \cdot (\boldsymbol{\xi} - \mathbf{m}_i(t))}{\|\mathbf{m}_i(t) + \alpha \cdot (\boldsymbol{\xi} - \mathbf{m}_i(t))\|_N} \quad (3)$$

where  $t$  is the discrete point at which events are sequentially sent to the network.

$\|\cdot\|_N$  is the normalization process, ensuring the weight vectors are bounded. The network's selectivity improves as more input vectors are presented. Network formation requires no initial correlation or order between any of the voxel input data elements. Hence, no initialization or operator assistance is required during feature cluster formation. Iterative steps lead array units to represent the topological order of different combinations of input image features.

### **1. Voxel Input Data Elements (VIDE)**

One dimensional feature maps with voxel intensity input data have been extended to include six (6) voxel input data elements (i.e., patterns) in a two dimensional processor and feature map for interactive surface segmentation. The image feature vector is generated for each input voxel, its elements are: (1) A distance from volume centroid to voxel location in three dimensional space; (2) The histogram equalized or normalized voxel intensity; (3) The Gaussian smoothed voxel intensity; (4) The difference of Gaussians; (5) The Laplacian zero crossings; and, (6) Invariant image moments.

The calculation of VIDE parameters is explained below. In general, intensity information is incorporated to provide voxel differentiation. Spatial distance parameters provide positional information, such as where a surface is located. Gaussian parameters used in self organizing map-based segmentation provide additional adaptive qualities to the 3D surface detection operation by smoothing artifact noise and compensating for scan protocol sequelae (resolution, pose, and beam strength). The "difference of Gaussian" parameter extends smoothing across a detected surface by introducing effects from local and distant voxels (i.e., the maximum distance between voxels is that of the surface detected but not elsewhere in the image volume). The "zero crossing" parameter provides transitional information important for some kinds of edge detection. The "moment invariant" parameters provide a positional and scale invariant signature of a surface in each region of the image volume; a computationally intensive parameter, moment invariants are initially set to 10 weight (Table I), and raised here only in small, user-defined, regions (e.g., the margins of thin structures) for fine tuning.

This set of voxel input data element vectors have normalized values of: spatial location via distance transform, voxel density with respect to nearest neighbors via Gaussian and difference of Gaussians, and boundary changes captured via Laplacian zero crossings. The two Gaussian input elements prevent feature inclusion of the spurious background noise commonly associated with 3D images. These parameters are computed during a preprocessing and feature-extraction procedure (see Figure 5). All VIDEs are normalized to a unit range of 0 to 1.0. This bounds the neural network output (i.e., second layer produced from VIDE input).

The voxel input data element vectors are defined as row vectors  $F_n = [E_n, V_n, g, g_d, Z_c, m]$  with six (6) elements, shown in Table I, where  $n$  varies from 1 to the total number of input image voxels. The weighting factors are defined for each element in order to prefer one element over the others. The resultant VIDE vector  $\xi$  is defined as:

$$\xi = \kappa F_n = [\kappa_1 E_n, \kappa_2 V_n, \kappa_3 g, \kappa_4 g_d, \kappa_5 Z_c, \kappa_6 m] \quad (4)$$

where  $\kappa_i$  defines the percentage of preference assigned for each VIDE. The differential preference of input VIDE results in associations between neural network processors and the voxel clusters. Table 1 presents the default preferences for each element during experiments.

5 **A. Input element 1: Normalized Euclidean Distance ( $E_{dn}$ )**

The radial distance from the image volume centroid is referred to as the moment of inertia. The image volume centroid is  $(C_x, C_y, C_z)$  as determined from:

$$\begin{aligned}
 C_x &= \left( \sum_{z=0}^d \sum_{y=0}^h \sum_{x=0}^w xV(x, y, z) \right) / \left( \sum_{z=0}^d \sum_{y=0}^h \sum_{x=0}^w x \right), \\
 C_y &= \left( \sum_{z=0}^d \sum_{y=0}^h \sum_{x=0}^w yV(x, y, z) \right) / \left( \sum_{z=0}^d \sum_{y=0}^h \sum_{x=0}^w y \right), \\
 C_z &= \left( \sum_{z=0}^d \sum_{y=0}^h \sum_{x=0}^w zV(x, y, z) \right) / \left( \sum_{z=0}^d \sum_{y=0}^h \sum_{x=0}^w z \right)
 \end{aligned} \tag{5}$$

where  $V(x, y, z)$  is voxel intensity,  $d, h, w$  are number of slices, slice height, and slice width. Euclidean distances are computed from the centroid to each input voxel as  $E_d = \sqrt{(x - C_x)^2 + (y - C_y)^2 + (z - C_z)^2}$ . This distance is normalized as

$$E_{dn} = E_d / \|E_d\|_{\max}.$$

15 **B. Input element 2: Normalized Voxel Density ( $V_n$ )**

This value depends on a prior choice of gray scale histogram normalization or equalization via a lookup table procedure. The value  $V_n = f(V(x, y, z)) / \|f(V)\|_{\max}$  is calculated from histogram mapping and normalized with the maximum value obtained from the entire volume. The best results are obtained via gray scale equalization.

20 **C. Input Element 3: Gaussian ( $\mathcal{G}$ )**

A 3D Gaussian is defined as:

$$G(x, y, z) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2+z^2)/(2\sigma^2)} \quad (6)$$

where  $\sigma$  defines the spread of the Gaussian curve as shown in Figure 6 . Use of two Gaussian features allows the neural network to produce feature clusters in noisy regions. Pre-computed numbers are used to normalize the 3D Gaussian. A  
 5 kernel is computed from this Gaussian function with pre-specified kernel size ranging from 2 to 10. The smoothed voxel intensity results from the convolution of the kernel over the present voxel and its immediate three (3) dimensional neighbors.

#### D. Input Element 4: The difference of Gaussians ( $g_d$ )

10 This is an extension of the above Gaussian input element with dual  $\sigma_1, \sigma_2$  values, shown separately in Figure 6. This is defined as:

$$G_d(x, y, z) = \|G_1(x, y, z) - G(x, y, z)\| \quad (7)$$

where  $G_1(x, y, z) = \frac{1}{2\pi\sigma_1^2} e^{-(x^2+y^2+z^2)/(2\sigma_1^2)}$  and  $G(x, y, z)$  is defined in equation 6.

As in Equation 6, pre-computed numbers are used to normalize this Gaussian  
 15 function. A kernel is computed from  $G_d(x, y, z)$  and  $g_d$  is (2x) obtained by convolving this kernel with the present voxel and its immediate neighbors (i.e., the 6 directly adjacent voxels).

#### E. Input Element 5: Laplacian Zero Crossing ( $Z_c$ )

Laplacian Zero Crossings are presented by as:

$$20 \quad Z_c(x, y, z) = \sqrt{(Z_c(x) + Z_c(y) + Z_c(z))} \quad (8)$$

this value is obtained from the Laplacian computed in each axis in the image volume, where:

$$Z_c(x) = \left( \left( \sum_{m=-d}^{-d} \sum_{n=-d}^{-d} V(x, y+n, z+m) \cdot \text{sgn}(n) \right) / L \right)^2,$$

$$Z_c(y) = \left( \left( \sum_{m=-d}^{-d} \sum_{n=-d}^{-d} V(x+p, y, z+m) \cdot \text{sgn}(n) \right) / L \right)^2,$$

$$Z_c(z) = \left( \left( \sum_{m=d}^{-d} \sum_{n=d}^{-d} V(x+p, y+n, z) \cdot \text{sgn}(n) \right) / L \right)^2$$

and, where  $\text{sgn}(n)$  is the sign of the scalar  $x$ .  $L$  is equal to  $(2d+1)d$  where  $d$  is a predefined length; and  $V$  is voxel intensity.

#### F. Input element 6: Moment invariant ( $m_i$ )

5 The moment invariant is an image statistic that is invariant to rotation, translation, and scale in a region. Moment invariants are derived from the definitions of moments, centralized moments, and normalized central moments. The image moments are defined as:

$$m_{pqr} = \sum_x \sum_y \sum_z x^p y^q z^r f(x, y, z) \quad (9)$$

10 where  $p, q, r \in \{0, 1, \dots, l\}$ . If  $f(x, y, z)$  is a piece-wise continuous function with bounded support in a 3D image, then the bounding is set by an  $m \times n \times o$  voxel array. It is positioned at the selected voxel and bounded by length  $l$ . The centralized moment  $\mu_{pqr}$  is defined as:

$$\mu_{pqr} = \sum_x \sum_y \sum_z (x - \bar{x})^p (y - \bar{y})^q (z - \bar{z})^r f(x, y, z) \quad (10)$$

15 where  $\bar{x} = m_{100} / m_{000}$ ,  $\bar{y} = m_{010} / m_{000}$ ,  $\bar{z} = m_{001} / m_{000}$  (i.e., centroid coordinates) and the point  $(x, y, z)$  is called the image region centroid. The normalized central moment is defined as:  $\eta_{pqr} = \mu_{pqr} / \mu_{000}^\gamma$  where  $\gamma = (p + q + r) / 2 + 1$ . From the normalized central moments, three (3) orthogonal invariant moments are defined as:

$$\begin{aligned} \phi_1 &= \eta_{200} \cdot \eta_{020} \cdot \eta_{002}, \\ \phi_2 &= 2\eta_{110} \cdot \eta_{011} \cdot \eta_{101}, \\ \phi_3 &= \eta_{020} \cdot \eta_{101}^2 + \eta_{011}^2 \cdot \eta_{200} + \eta_{110}^2 \cdot \eta_{002} \end{aligned} \quad (11)$$

25 These three orthogonal moment invariants were selected because they are irreducible and distinct. When approaching tissue boundaries the moment invariants may provide information that complements the zero crossing data. The

strength of the zero crossing 3D surface signal is expected to be more dependent on the object's pose (i.e., relation of surface orientation to scan orientation) than the moment invariants.

## **2. Back projection of Feature Clusters**

5 In the preferred embodiment, the SOFM neural network utilizes 4x4 to 16x16 processor network configurations in either a rectangular or a hexagonal grid formation (see Figure 7). In a rectangular grid, each processor has four connections to its neighbors. In a hexagonal grid, each processor unit has six connections to its neighbors. The input layer, translates each voxel into a vector  
10 containing the VIDEs defined in Equation 4. The input layer has direct connection to all processors in the formative network configuration and is associated with a weight as shown in Figure 5. A flowchart of the complete surface segmentation process is shown in Figure 8.

### **A. Preprocessing to Remove Background Noise and Head Holder**

15 The head holder and background noise are separated from the head object in a two step procedure. First, a grayscale threshold is manually set; next, a volumetric region growing method is applied. This removes voxels associated with the head holder, background noise, and most metallic artifact outside the head; it also limits the voxels sent to the neural network to those relevant to the 3D  
20 surfaces internal to the head.

### **B. Randomization of Input Voxels**

The randomization of the VIDE vectors in the input event space is necessary in order to obtain an ordered map. Each voxel intensity in the image volume is stored along with its position (i.e., x, y, z coordinates) in a randomly  
25 selected entry in a hash table (i.e., a register queue length  $N$ ). The random position  $R_p$  in the hash table is computed as  $R_p = \text{hashTable}[\text{length} \times \text{randNumber}]$ , where the hash table is initialized sequentially from 0 to the total number of voxels  $N$ . The random number is generated in a range between 0.0 and 1.0. The length of the hash table starts with  
30 value  $N$ . Subsequently,  $N$  decrements for each position computed and is

swapped with entries for that position's current length in the hash table. This procedure ensures randomization of feature vectors in the input element event space and, subsequently, the map formation by adapting weights in the self organization process. The position of the intensity in the hash table entry is replaced during map formation with the processor identity; processors represent feature clusters of potential anatomical surfaces.

### C. Self-Organizing Feature Map Algorithm

With reference to FIGURES 8 and 9, weights are randomly initialized around  $0.5 \pm x, x \in \{0.0, \dots, 0.1\}$ , in a step K, with default initial learning rate and neighborhood excitation parameters, set by the user.

A pattern  $\xi$ , a VIDE vector, is presented, and the network output is evaluated in a step L.

The unit  $\{i, j\}$  with the minimum output is selected and designate as a winner:  $|P_{i,j}|_{winner} = \|\xi - W_{i,j}\|_{min}$  in a step M.

Weights are updated, in a step N, using a learning rule where neighbor excitement levels are a function of self and neighbor excitation rates

$$|W_{i,j}(t+1)|_{winner} = W_{i,j}(t) + \alpha_{i,j}(t) \cdot [\xi - W_{i,j}(t)],$$

$$W_{m,n}(t+1) = \begin{cases} W_{m,n}(t) + \beta_{i,j}(t) \cdot [\xi - W_{i,j}(t)], & \text{if } ((m,n) \in N_{i,j}(t)) \\ W_{m,n}(t) & \end{cases},$$

where  $N_{i,j}(t)$  is a neighborhood of  $\{i, j\}$  at time  $t$ ;  $\alpha_{i,j}(t)$ , the learning rate for the winner processor;  $\beta_{i,j}(t)$ , is the rate at which its excitation influences its neighbors.

The value of  $\alpha_{i,j}(t)$  and  $\beta_{i,j}(t)$  is decreased, and the neighborhood,  $N_{i,j}(t)$  is shrunk in a step O.

Steps K-O are repeated for all the VIDE vectors:

$$W_{i,j} = [w_1, w_2, w_3 \dots w_6] \quad i = 1, 2, \dots, I \quad j = 1, 2, \dots, I \quad I \times I \quad I \in \{4 \dots 16\} \quad Q_{i,j} = \|\kappa F'_n - W_{i,j}\|$$

$$\xi = \kappa F'_n$$

$$n \in \{1 \dots N\} \quad N \quad Q_{i,j} \quad \alpha_{i,j}(t) \quad \alpha_{i,j}(t) \quad \beta_{i,j}(t) \quad \beta_{i,j}(t) \quad \gamma_{i,j} \quad id \in \{0 \dots 255\} \quad W_{i,j} \quad N.$$

### 3. Interactive Image Segmentation and Rendering

The SOFM 3D surface segmentation program integrates the SOFM algorithm, volumetric image reading, slice image display and editing, and graphical surface rendering of output feature cluster voxel data (i.e., obtained 3D surfaces).

The 2D SOFM processor map displayed in the window (Figures 10, 11, and 12) allows the user to select voxel clusters associated with each processor. Clicking on a displayed node activates back projection of the associated voxels to a 3D graphics window. Successive feature clusters associated with desired surfaces may be accumulated, saved as a group, and back-projected to the graphics display (Figure 10). At this stage (i.e., selection of map nodes) no surface rendering is provided.

Our tests were with 3D head CT data. The voxel display of these data allows the user to quickly identify the processor nodes associated with skull, soft tissue face, oral, or cervical vertebral surfaces. After having identified the group of processors associated with the desired 3D surface, the voxels are triangulated for 3D surface rendering.

#### A. Self Organization (Ordering) Process

During the automated image segmentation process an ordering, from an initially random configuration of processors to a state of equilibrium of the self organizing feature map must occur. The weight vector of a processor is given by  $W_{i,j} = [w_1, w_2, w_3 \dots w_6]$ , where  $i = 1, 2, \dots, I$ ,  $j = 1, 2, \dots, I$ , and  $I \times I$ , is the number of processors assigned to the network configuration, where  $I \in \{4 \dots 16\}$ . A maximum of 256 processors were chosen, a condition which insures that the identity of a processor is always found in the range of 0 to 255. This allows each processor to be stored within one byte, thereby saving memory for further computation.

A VIDE is generated from each voxel in the input volume event space hash table and is sent to the processor layer. Once a processor captures a



voxel, the cluster identity is placed back in the hash table register replacing the input voxel intensity cell datum. Later it will be displayed as part of a surface with all voxels belonging to this cluster. Voxel intensity information is obtained from the original input image volume via corresponding hash table registers. A processor's computed output is:

$$Q_{i,j} = \|\kappa F'_n - W_{i,j}\| \quad (12)$$

where  $\xi = \kappa F'_n$  is a VIDE vector,  $n \in \{1 \dots N\}$  and  $N$  is the total number of voxels.

Among the processors a winner is identified. The distance measure  $Q_{i,j}$  determines if it is lower than the initially chosen value. If so, the processor identity changes from its initial state; the new processor will continue to claim the voxel as long as no other processor claims it is the "winner." The winner successively strengthens its weight vector by a fraction of the input element values sampled by the  $\alpha_{i,j}(t)$ , a self excitation parameter. The value of  $\alpha_{i,j}(t)$  is set and then decreased successively, eventually resulting in no change to the weight vector. This ensures that the network will reach an equilibrium. Another parameter,  $\beta_{i,j}(t)$ , influences neighborhood processor excitation and determines how much the winner can excite or inhibit its neighbor's weights.  $\beta_{i,j}(t)$  is assigned a fixed value that decreases over time to avoid wide fluctuations in the organization process. Using neighborhood Gaussian function curves (See Figure 3), defined by the parameter  $\gamma_{i,j}$ , the number of neighboring processors is successively decreased. This number is initialized with a default value at the beginning. The "winner" processor takes the current input voxel to its cluster along with its spatial coordinates. This update replaces the processor's identity,  $id \in \{0..255\}$ , with the voxel density currently in the register.

Prior to the mapping process the weight vectors,  $W_{i,j}$  of each processor, are initialized with random values. Therefore, the initial topology of the axis spanned by connection strength between any two of the VIDE elements is unorganized. This is shown in Figure 13A. The processor's physical connections

remain constant throughout the process, (i.e., processor 1's connections to processor 2 remain the same). However, the mapping topology with respect to each processor's connection strength changes throughout this process. The random starting topology, shown in Figure 13A, is drawn by plotting each processor's weight vector, chosen two weight elements at a time, onto two dimensional axes.

5 The initial pattern has no meaning, however the final pattern obtained presents processors (feature clusters) containing voxels representing 3D surfaces and similar 3D surfaces will be shown in the same SOFM map region. We have chosen voxel intensity and Euclidean distance and their corresponding members in processor weight vectors to plot the map topology in Figures 13, 10, and 11.

10

After  $N$  input VIDEs are fed through the input layer, producing each processor's final weight vector, the network adopts, approximately, the original "flattened" processor grid topology. The axis of the grid is spanned by processor connection strength values. This grid is a 2D projection image of the feature cluster VIDE hyperspace. An example is seen in Figure 13B.

15

SOFM feature vector adaptation results from two basic mechanisms: (1) Identifying a "winner" and allowing it to modify its connection strengths; (2) allowing the winner to selectively excite/inhibit its neighbors. The first mechanism ensures connection strengths that reflect input event values. The second mechanism ensures that related processors, especially those representing anatomical surfaces, are mapped adjacently. The observed surfaces, displayed as point clusters from a set of neighboring processor nodes, illustrate the second mechanism in Figure 10.

20

The voxels associated with the winners are clustered and classified as separate groups. These clusters are identified through the processor's identity, a unique value which is stored in the hash table during the self organization process. The resulting clusters capture continuous anatomical surfaces [Figure 10] that exist in the 3D image volume as an intrinsic model of the VIDE event space. This approach differs from physically based models (prior information) or user-defined (algorithm assisted or manual selection) segmentation methods that rely on information extrinsic to the image volume. Our method adaptively locates surfaces

25

30

captured in the image volume using voxel data intrinsic to the image volume. We involve the user to verify and determine the surface identity. This introduces supervised learning to the unsupervised neural network algorithm.

## **B. Surface Rendering**

5 Two different types of surface rendering methods (i.e., triangulation of voxels associated with selected feature clusters) have been implemented. The first approach is primarily a contouring of the outermost voxels. Assuming the volume contains a thin boundary surface, all cells on the surface are classified as useful vertices. Each slice is divided into four quadrants, and each quadrant is  
10 swept by 256 lines from it's outer most volume boundary to the slice image centroid. In each ray, the first encounter with the boundary surface is defined as a boundary vertex. The remaining sweeps obtain subsequent vertices in each slice, resulting in a closed curve that traverses all of the detected slice vertices. Since the line sweeps are produced at half degree angular rotation intervals, the points  
15 generated are aligned in the correct neighborhood. Next, four sided polygons are generated from all quadruples of neighboring vertices in a pair of consecutive closed curves. These polygons are degenerated to two triangular faces. This results in a complete triangulated surface of the segmented image object. This procedure is extremely fast and well suited to display simple surfaces such as the  
20 soft tissue face (see Figure 12A). However, views of objects with complex topology, such as the skull, provide unsatisfactory results.

A second approach is taken to rendering complex surfaces such as the skull. We used a known method of decomposing volumetric data into tetrahedra. This approach takes two adjacent slices and traverses neighboring  
25 voxels eight at a time, treating them as a tetrahedron. The decomposition of tetrahedra produces the oriented cycles necessary to represent triangular faces of an isosurface. This approach yields, for example, a skull surface (see Figure 12B). We also compared the Doi and Koide tetrahedral decomposition to the marching cubes algorithm. Both algorithms use bi-linear interpolation to compute partial  
30 voxel values. Computation of a complete skull surface rendering took

approximately 3 to 4 minutes as opposed to 2 to 3 minutes, on an SGI Octane workstation.

### C. Combining Multiple Surfaces and Filling Holes

The user can combine the feature clusters associated with multiple SOFM processors. If unexpected holes are seen in the surface, perhaps due to partial voluming "drop out" of thin structures, the volume image region immediately around the missing area may be selected as a new region of interest (ROI). A new high resolution processor map of voxel clusters is generated for this zoomed-in ROI, allowing careful surface assembly in regions of thin bone or skin.

### 10 D. Partial Volume Error

This error results in surface "drop out" and is a well known cause of surface inaccuracy in 3D CT visualization of thin bone segments. The problem becomes more difficult where bone voxel intensities are reduced to tissue levels. Region growing and manual intensity thresholding techniques for isosurface generation cannot overcome these errors. We find that selecting a region presenting thin bone segments and assigning higher preferences to VIDEs 5 and 6 (i.e., Laplacian zero crossing and Invariant image moments) allows a local SOFM map to capture more of the desired voxels representing surfaces formed by thin bone. For example, the computationally expensive moment invariant was set at 10 weighting on this initial pass, then elevated to 40-60% in user-defined regions around thin bone structures such as the infra- orbital foramina and orbital walls (Figure 14).

## 4. SOFM Precision Tests and Results

The SOFM program was tested with five 3D CT data sets (B2537, B2621, B3037, B3095, B3195) provided by a recall of subjects. All subjects are above 60 years in age, female, have dental fillings, and often have large oral prostheses (dentures). One operator (AM) segmented all the images twice by two methods: (1) slice-based methods (i.e., intensity thresholding, manual pixel selection, and 2D and 3D region growing) in the AUTO program, and (2) Self Organizing Feature Map. Both the slice-based and SOFM tools provide orthogonally reformatted views of the original image and segmented pixels. More

than a day passed between each segmentation session. Intra-operator variance (precision), using either segmentation method, was measured between the two segmentation attempts. In manual segmentation, skull and soft tissue face segmentations were separate activities. In SOFM experiments, the same ordered  
5 map was used to produce the skull and soft tissue surfaces in a single attempt.

The original, unsegmented CT data sets are 512x512x115 in dimension with 1.904 mm slice thickness and 0.5mm pixel resolution. This scan protocol was chosen to provide accurate source data for boney skull and soft tissue face isosurface images. This data was reduced to 256x256x115 in dimension with  
10 1.904 mm slice thickness and 1 mm pixel resolution. This resolution is sufficient in terms of the resulting surface segmentation and saves memory resources on the computer.

Five measures were used to determine both intra-operator within-method variance and between-method variance in the resulting segmented images.  
15 The former measures how reproducible the segmentation is; low variance implies high precision. The latter measures how different the results of the two methods are. We expect the differences to be local to regions of thin structures or metallic artifact where qualitative inspection suggests the greatest benefit from SOFM segmentation. These five measures are: (1) Visual comparison of the skull and soft  
20 tissue face surfaces obtained; (2) Difference of the overall volume of the segmented image; (3) Two dimensional difference images computed slice by slice for each slice image within the volume; (4) Three dimensional difference images viewed from three orthogonal eye points; and (5) Procrustes superimposition of  
25 manually located, highly reliable, single point anatomical landmarks. The last measure is similar to a qualitative visual determination of the completeness of anatomical features seen on the segmented surface. Clearer features will result in more easily, and more repeatably, detected anatomical landmark coordinates by trained workers.

**A. Test 1: Visual Comparisons:** The left hand side of Figures 14 and 15  
30 show both manual slice-based segmentation attempts. The right hand side shows both SOFM segmentation results. SOFM segmentation of soft tissue face recovers

thin tissue structures around the nostrils, lips, and ears, producing an improved image. There is also dramatic improvement in reducing the effect of the metallic (dental filling) artifact around the lips. Note the variable posterior sagging of inframandibular soft tissue due to the affects of gravity on the supine individual within the head holder. SOFM skull segmentation (Figure 14) improves areas of thin bone around the infraorbital and mental foramina and the thin posterior and medial orbital walls. These visual results indicate that the slice-based segmentation does not allow as effective a search for these surfaces.

**B. Test 2: Volume Differences:** Table II presents volumetric differences between the soft tissue face obtained between an average of multiple attempts and each attempt in both manual and SOFM methods and between the two method averages. Table III presents the same data for the segmented skull image volumes. The difference image volumes were computed by multiplying the total number of voxels in each segmented image by voxel volume (1.904 mm<sup>3</sup>). SOFM segmentation results show less difference between sessions. Given the additional detail it is not surprising that in comparison with manual slice-based segmentation, SOFM segmentation results in an average of 3.9% more volume for skull segmentation and 0.5% more for soft tissue face segmentation. The higher difference for skull segmentation likely indicates greater reduction of partial volume error.

**C. Test 3: 2D Slice Image Difference Data:** This test presents 2D slice by slice difference profiles for each image volume using either segmentation method. These difference profiles result from calculating the absolute difference between corresponding pixels in the homologous segmented slices. The accumulated difference across all five data sets are plotted per slice in Figure 16A for the soft tissue face surface and Figure 16B for the skull surface. Regions where the slices contain thin bone and dental filling artifact require manual pixel selection during manual slice-based segmentation in AUTO. These are areas where the user must make many decisions that are not well replicated on subsequent segmentation attempts. Manual slice-based segmentation results in higher intra-operator variability. Regions with thin soft tissue or bone, or dental artifact, contributed

most to the difference plots (Figure 16) between the slice-based manual and SOFM segmentation results, this suggests that the latter was more the reliable method.

**D. Test 4: 3D Difference Images:** This test consists of 3D difference images computed between the volume image resulting from the two segmentation methods. The 3D difference images, seen in three orthogonal projections of 2D images in Figure 17, graphically illustrate the affect of highly variable human decision making in the regions of metallic artifact and thin structures during manual slice-based segmentation. Note that thin areas of the skull surround the pneumatic sinuses (i.e., frontal, maxillary, ethmoid, sphenoid, and mastoid). Areas of thin soft tissue face structures (especially, the nostrils, eyelids, ears, and lips) are more fully extracted by SOFM than with slice-based manual methods.

**E. Test 5: Anatomic Landmark Localization:** The user was asked to pick reliable anatomic landmarks twice on the soft tissue face and skull surface segmented either manually or by SOFM. These landmarks include 21 on the soft tissue face surface (Table IV) and 26 on the skull surface (Table V). The error found between landmark locations in these two sessions was determined via Procrustes fit. The error reported here is the average value across all 5 superimpositions. We contend that the improved quality of the imaged surface, especially in areas of artifact or thin structures, reduces intra-operator variance (i.e., raises precision) between landmarks picked on two attempts of either the soft tissue face or skull surfaces segmented via the SOFM method; the less ambiguous SOFM surface renderings allow our skilled operator to place these landmarks more precisely.

**F. Time of Segmentation:** Skull and soft tissue face segmentation sessions (AM) were reduced from approximately 3 to approximately 0.5 hours. In either method, the majority of time is spent correcting dental artifact or partial volume error. We are aware that the skull is most often segmented at the CT console by intensity thresholding alone. This method may save time but results in the poor segmentation of thin structures or those near metallic artifact.

## **II. Warping**

As described below, the step **G** of warping the template to an external surface of a normative shape of the bone of interest is preferably executed using a Simulated Annealing-based Surface Extraction (SASE) process.

5 In order to save time, improve precision (repeatability), and increase accuracy of the extracted anatomical surface, a ridge-curve based deformable template superimposition strategy combines simulated annealing of a ridge and tiling curve wireframe and subsequent surface normal based identification of surface tile points. The annealing algorithm searches the volume image and  
10 displays the result to the operator on a graphical manifold rendering of the surface of interest. Unlike energy minimization algorithms, which require a good initial parameter estimation to avoid converging to a misleading local minimum, annealing methods exploit stochastic search methods to locate a global minimum. Several stochastic methods form a basis for simulated annealing. Space curve  
15 fitting of 3D surfaces is a goal not uncommon in the field of computer vision. It is merely object recognition followed by shape registration.

B-spline space curve encoding of labeled surfaces has application beyond surface averaging and craniometrics or image co-registration. We seek highly accurate surface extractions to model cranial prosthetic implants for the  
20 skull. Homology mapped, B-spline space curve encoded surfaces prove useful in morphometric, biomechanical, and deformable model assessment of many organ systems. Additionally, parameterized surface extractions, especially of the external body surfaces, have application in animation.

### **1. Ridge Curve-based Deformable Template Creation**

25 We chose to use B-spline surface encoding because it provides a smooth surface approximation, reduced data set, and easier assignment of labels to various components (see Appendix 2 for B-spline curve and surface construction). We have re-implemented skull and soft tissue face ridge curve-based deformable templates in this fashion. Figure 18 shows one of these new ridge curve-based  
30 deformable template wireframes superimposed on a soft tissue face graphical manifold, with Type II landmarks, ridge curve arcs, and tiling curves indicated.



Tiling Curves: Since ridge curves rarely intersect, additional space curves are necessary to tile the surface with a single enclosing wireframe. In order to identify a unique demarcating line within a search space we define tiling curves as space curves that: (a) traverse a low curvature area (i.e., flat surface); and (b) are weighted to pass close to a plane passing through the surface between the tiling curve end points and the center of mass of the entire image surface (See Sections II.2.C and II.2.G).

Initially, we approximate tiling curves as straight lines between endpoint landmarks. In Figure 18, B-spline ridge and tiling curves are indicated by dotted lines while their control points are shown as small black cross-hatches. The Type II landmarks are shown as large black cross-hatches.

As one draws either a ridge or tiling curve, a corresponding B-spline curve is fit to the surface via global curve interpolation to a set of control points sampled from the user drawn space curve. The B-spline interpolated 3D points collected are re-sampled and indexed to vertices on the graphical manifold surface and assigned a control point vector,  $P_i$ . In the case of ridge curves, our re-sampling maintains the curvature information along ridge curves when determining how to space the control points. On the other hand, tiling curve control points are equi-distantly spaced. The curve smoothness, set by the operator, further determines redistribution of control points, if necessary. Knot-vectors and knot-spacing require a B-spline interpolation function assignment for each space curve, by the operator during the template creation process.

Figure 19A and 19B display the Type II landmarks and Appendix 1A and 1B list the non-landmark components of the soft tissue face and boney skull templates, respectively. The soft tissue face template consists of 56 Type II landmarks, 103 space curves (54 ridge curves and 49 tiling curves), and 44 surface tiles [Figure 19A, Table VI]. The skull template consists of 54 Type II landmarks, 104 space curves (40 ridge curves and 64 tiling curves), and 47 surface tiles [Figure 19B, Table VII].

## **2. Ridge Curve-based Deformable Template Superimposition**

**A. Graphical Surface and Volume Image Input:** The 3D surface input we use results from segmented volume images. The segmented voxels are rendered as a graphical manifold. Segmentation occurs via an unsupervised neural network program. The graphical manifold is built with sub-sampled voxels to provide an interactive context to the operator.

The simulated annealing ridge curve-based deformable template superimposition algorithm evaluates features found in the volume image, not on the graphical manifold surface. The unsubsampled volume image provides additional cues for best locating ridge or tiling curves. The result of the search is displayed on the graphical manifold surface where the operator can make manual corrections to algorithmic insufficiencies. The CT slice shown in Figure 20A highlights the segmented surface voxels of interest which are indexed to the vertices (points) in the adjacent rendered manifold surface (Figure 20B).

During loading of the graphical manifold surface data, a 3D lookup table (i.e., geometric hash-table) of spatial  $x, y, z$  and curvature metrics (Listed in Table 8) corresponding to each vertex is generated. This provides the simulated annealing based surface extraction complete indexing between the rendered graphical manifold and the source volume image.

**B. Candidate Wireframe:** The first step in the Simulated Annealing-based Surface Extraction (SASE) process is the operator's manual location of the Type II landmarks on the graphical manifold surface. These landmarks attach the ridge curve-based deformable template to the graphical manifold surface via a thin plate spline warp. Figure 21A shows a candidate wireframe prior to the warp at lower left. Figure 21B shows the result of the thin plate spline warp of the wireframe onto the Type II landmarks. Some of the landmarks (large cross-hatches) are visible. Note how the candidate wireframe undulates in and out of the surface.

**C. Volume Image Features:** Simultaneous to loading of the graphical manifold data and lookup table generation, a set of surface geometry features are generated for use in the simulated annealing procedures. A method is known for computing principal directions and principal curvature values (Figure 22) for any

point on the segmented image surface directly from the input volume image data. In addition, the principal normal and tangent vectors are derived at the same time. These latter parameters, in addition to providing the basis for the simulated annealing process cost functions, are also used to bound the search space in the volume image linked to the graphical manifold surface displayed for the operator.

The search for volume image principal curvature normal vectors and principal curvature tangent vectors occurs initially at a point,  $I$ , a control point found on the B-spline space curve, defined at location  $\{x_0, y_0, z_0\}$  to have normal vector  $\bar{N}$  on the volume image surface  $S$  defined as  $I(x, y, z) \equiv I(x_0, y_0, z_0)$ .  $I$  lies in the same direction as the curvature gradient (i.e., changing surface normals) of the surface voxel data:

$$\nabla I = (I_x, I_y, I_z) \mid (\bar{N} = (\nabla I) / \|\nabla I\|) \quad \dots(\text{EQ } 13)$$

In order to evaluate the local surface curvature we consider a space curve  $C$  from a sweeping plane positioned at each control point, with tangent vector  $\bar{t}$ , along a normal section of segmented voxel surface  $S$  (Figure 22A). The normal vector at any point along  $C$  is  $N = (n_x, n_y, n_z)$ . Differentiation of  $\nabla I \cdot t = 0$  yields surface normal curvature  $k_n$  in direction  $\bar{t}$ , where  $k_n$  attains principal curvature values  $k_1$  and  $k_2$ . Principal directions  $\bar{e}_1$  and  $\bar{e}_2$  show the path of the principal curvatures (Figure 22B).

On any surface, the local curvatures vary between the principal curvatures  $k_1$  and  $k_2$ . Therefore, the curvature parameters we use combine these measures:

$$K = k_1 \cdot k_2 \quad \dots(\text{EQ } 14)$$

$$M = \frac{1}{2} \{k_1 + k_2\} \quad \dots(\text{EQ } 15)$$

where  $K$  is Gaussian curvature, independent of principal directions and  $M$  is mean curvature changes sign with opposing principal directions.

Twelve feature dimensions are used for simulated annealing registration of ridge or tiling curves (Table VIII). The 3rd order image derivatives are obtained from image derivatives:

$$H = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yz} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad \dots(\text{EQ 16})$$

5 The Partial derivatives of  $H$  along each axis are  $H_x, H_y, H_z$ :

$$H_x = \begin{bmatrix} I_{xxx} & I_{xyx} & I_{xzx} \\ I_{yxx} & I_{yyx} & I_{yzx} \\ I_{zxx} & I_{zyx} & I_{zzx} \end{bmatrix}, H_y = \begin{bmatrix} I_{xxy} & I_{xyy} & I_{xzy} \\ I_{yxy} & I_{yyy} & I_{yzy} \\ I_{zxy} & I_{zyy} & I_{zzx} \end{bmatrix}, H_z = \begin{bmatrix} I_{xxz} & I_{xyz} & I_{xzz} \\ I_{yxz} & I_{yyz} & I_{yzz} \\ I_{zxz} & I_{zyz} & I_{zzz} \end{bmatrix} \dots(\text{EQ 17})$$

Among these 27 third order partial derivatives, ten derivatives are chosen,  $I_{xxx}, I_{xxy}, I_{xxz}, I_{xyy}, I_{xyz}, I_{xzz}, I_{yyy}, I_{yyz}, I_{yzz}, I_{zzz}$ , preferring derivatives along the three axes. Essentially, the maxima of these ten third order partial derivatives provide an indication of sharp geometric transition. A pre-determined inter-knot spacing restricts any overlap of search space between any two control points. The Type II landmarks are used for ridge curve arc-separation and to anchor tiling curves.

15 Features for Ridge Curve Arcs: We use Gaussian and mean curvatures, and ten third order partial image derivatives to obtain maximum curvature prescribing control points along the surface between ridge curve arc endpoints (i.e., Type II landmarks).

20 Features for Tiling Curves: We use the same set of features to position tiling curve control points along lines of minimum curvature. In addition, we use a thirteenth feature, a deviation measure,  $d_m$ . It is found between the candidate control point normal and the nearest ray within a plane constructed from the two endpoint landmarks and the object's center of mass (Figure 23). The deviation measure helps limit the tiling curve search space so that no boundaries overlap.

25 **D. Initial Control Point Location:** When the template is warped to operator located landmarks on a patient image the candidate space curves connecting the

Type II landmarks and their control points may not lie on the image surface. The true surface is located using a trace along the candidate control point normal. Thus, the initial candidate control point is assured to be on the image surface. Figure 24 illustrates a trace along the candidate template's control point to the  
5 image surface. The annealing process starts with the first point found on the surface as an initial candidate control point.

**E. Search Plane:** A search plane is anchored to the candidate wireframe (i.e., ridge curve-based deformable template) and rotates about the axis of that point's surface normal in the volume image. Note that when ridge or tiling curves sit on a  
10 two sided tile boundary, these cut planes rotate a full 360 degrees (Figure 25B). The sweep rotation about each control point normal is set to increment a default 45 degrees (operator chooses between 10 and 90 degrees). If search in one direction fails, an opposite direction search is attempted. This is important where a ridge arc is at the surface margin.

**F. Random Candidate Control Point Generation:** The principal directions constrain the search space for each control point along the B-spline space curves. The maximum principal curvature bounds the space in which it is profitable to search for optimal B-spline space curve control points. The search space for each B-spline control point during ridge curve or tiling curve superimposition is defined  
20 from the point normals and tangents bounded by the principal directions. Figure 25A shows a cross section on the top of the skull and tiling curve control point search planes intersecting that image surface. Figure 25B shows a series of search plane iterations.

Candidate control points are evaluated during each search plane  
25 iteration. Search plane rotations select a randomly located set of  $M$  (chosen by operator, default is 2) candidate control points for  $k$  search iterations. Random candidate positions are selected from the space curves intersected by the search plane rotations through the image surface. Their features (Table VIII) are computed and evaluated by the cost function in comparison to the current  
30 candidate control point position,  $\zeta$ . The actual candidate locations on the image

surface chosen for evaluation via variables  $r$  and  $s$  using reflection coefficient parameter  $\alpha$  :

$$r = \left( \frac{\alpha}{N\sqrt{2}} \right) \cdot (\sqrt{(N+1)} + N - 1.0) \quad \dots(\text{EQ 18})$$

$$s = \left( \frac{\alpha}{N\sqrt{2}} \right) \cdot (\sqrt{(N+1)} - 1.0) \quad \dots(\text{EQ 19})$$

where  $N$  is the number of dimensions (i.e., number of image features in the case presented). Randomly distributed position vectors ( $I_p$ ) are computed by using  $r$  and  $s$  as:

$$I_p = S(h \cdot \text{rand}(r), w \cdot \text{rand}(s)) \quad \dots(\text{EQ 20})$$

where  $S$  is the search space spanned by width,  $w$ , and height,  $h$ , computed from the normal and tangent at the template B-spline space curve control point. The  $\text{rand}(r), \text{rand}(s)$  are the coordinates with the search plane. Each position within the search space plane carries a single vector for the three coordinate axes, which is indexed to a segmented surface voxel in the volume image space.

The probable distribution of positions are subsequently used in finding the best ridge or tiling curve control point candidate. The values for  $r$  and  $s$  are decreased after each iteration:

$$r = r - (r/k), s = s - (s/k) \quad \dots(\text{EQ 21})$$

and successively decreased, as  $k$ , the current iteration, approaches the maximum number of iterations,  $k_{\max}$ .

**G. Simulated Annealing Cost Functions:** The cost function evaluates the candidates for B-spline control points intersected by the search plane via simulated annealing. The procedure used here is based on the Metropolis algorithm.

Ridge Curve Cost Function: The value of the ridge curve search space cost function is computed from the twelve features (Table VIII):

$$E(\zeta) = \sum_{feature_i=0}^{12} F_{\zeta_i}^{-2} \quad \dots(EQ 22)$$

where  $\zeta$  is a candidate space curve control point. Since the ridge curve requires maximum curvature values, the inverse of these squared distance feature is minimized during simulated annealing.

5                    Tiling Curve Cost Function: For tiling curves, the cost function is defined as:

$$E(\zeta) = \sum_{feature_i=0}^{13} F_{\zeta_i}^{-2} \quad \dots(EQ 23)$$

The curvature-based features, as well as the deviation measure ( $d_m$ ) are minimized directly as squared distance features for tiling curve control point identification  
10 (Table VIII).

**H. Illustration of Control Point Location:** The rotating search planes are intended to identify optimal control point locations to assign to the ridge or tiling curve (Figure 26A). For example, search planes for one ridge curve (i.e., menton\_to\_l\_gonion) are arrayed along the warped ridge curve-based deformable  
15 template space curve in Figure 26B. Here these search planes pass along the candidate ridge curve control point normal superiorly to find intersection with the voxels representing the base of the mandible. Figure 26C, shows the final ridge curve registration found by the simulated annealing algorithm. Similarly, Figure 26D shows a candidate tiling curve (i.e., r\_ASSZ\_to\_r\_PBP) and its associated  
20 search planes. Figure 26E shows the fitted control points. Figure 27 shows the entire final ridge and tiling curve wireframe superimposed on the image surface.

**I. Simulated Annealing Algorithm Description:** Four key elements are considered in our simulated annealing algorithm implementation: 1) The configuration definition; 2) Random candidate generation mechanism (i.e.,  
25 definition of neighborhoods on the configuration space); 3) A cost function; and 4) A cooling schedule. In our implementation, we have first defined the configuration space with respect to the curvature features necessary to locate ridge or tiling curves on an image surface (Section II.2.C). Second, candidate control

points are randomly generated from the search planes (Section II.2.F). Third, standard cost functions for location of ridge and tiling curves have been defined (Section II.2.G). Finally, the annealing procedure is described here.

The annealing algorithm (Figure 28) attempts to minimize the cost function  $E(\zeta)$  where  $\zeta$  is a continuously modified candidate B-spline space curve control point, which lies in a search plane intersecting the surface. A set of cost functions  $E_j(\zeta)$  are constructed from the sequence of template control point deformations  $\zeta^1, \zeta^2, \dots, \zeta^k$  starting from  $\zeta^0$  at the initial ridge curve-based deformable template control point estimate. For each deformation, we have a set of  $P$  candidates  $\zeta_0^k, \zeta_1^k, \dots, \zeta_p^k$  within multiple search plane in current iteration,  $k$ . From the initial candidates, the best candidate  $\hat{\zeta}^k$  is selected.

$$\hat{\zeta}^k = \zeta \text{ of } \|E_j(\zeta^k)\|_{\min} \dots (\text{EQ 24})$$

as  $k$  approaches the maximum number of iterations the cost function  $E(\zeta)$  reaches a stable control point  $\zeta^s$ . Simultaneously, the temperature,  $T_k$ , a component of the annealing procedure, is set to decrease monotonically. This parameter limits the search area and determines the probability of determining the best candidate between iterations. As the temperature lowers, the search area and number of probable candidates, narrows. At each iteration,  $k$ , the algorithm determines a new value of the B-spline control point deformation parameters, based on the previous value (i.e.,  $\zeta^{k-1}$ ) and compares it with the present value.

$\zeta^k$  is determined with probability  $p_k - 1$ , which are defined as:

$$\zeta^k = \begin{cases} \hat{\zeta} \in p_k - 1 \\ \zeta_{k-1} \notin p_{k-1} \dots \end{cases} (\text{EQ 25})$$

$$p_k - 1 = \min \left( \exp \left\{ \frac{-[E(\hat{\zeta}) - E(\zeta^{k-1})]}{T_{k-1}} \right\}, 1 \right) \dots (\text{EQ 26})$$



$T_k$  is a monotonically decreasing temperature sequence,  $E(\hat{\zeta})$  is the cost function computed at the current position  $I$ , and  $E(\zeta^{k-1})$  is the cost function computed in the preceding iteration.

The temperature schedule  $T_k$  [34]:

$$T_k = T_0 \left( \frac{T_f}{T_0} \right)^{\frac{k}{k_{\max}}} \dots \text{(EQ 27)}$$

5

where  $T_0$  is initial temperature,  $T_f$  is final temperature, and  $K_{\max}$  is the number of maximum iterations allowed. Figure 29 presents the temperature cooling schedule from the initial temperature 100 to the final temperature, -10. We set the number of iterations to 120 for our initial testing (see Section 4).

- 10 **J. Final B-spline Ridge and Tiling Curve Registration:** Figure 30 presents an overview of the ridge curve-based deformable template registration process in SASE. Manual editing of control point location is provided in the event that the simulated annealing algorithm fails to register a ridge curve arc to the operator's satisfaction. The operator may also locally vary the search parameters (i.e., sweep  
15 rotation degree, search plane height and width, or annealing temperature).

### 3. Surface Extraction

- The registered B-spline control points are interpolated to produce a set of tile boundary points (i.e., smooth ridge or tiling curves). We set the default number of tile boundary points for each space curve to 50, assuming this would be  
20 adequate in all cases. Each tile, therefore, consists of 2500 parametric points. Our soft tissue face ridge curve-based deformable template consists of 44 surface tiles [Tables VI and VII], 106376 tile points. Our skull ridge curve-based deformable template consists of 47 surface tiles [Tables VI and VII], 113334 tile points. Note that some of the tiles are constructed with three bounding curves and the fourth  
25 side is a degenerate endpoint of the first and third sides. Since they are redundant, we can reduce the number of tile points encoded somewhat from the fully quadrilateral tile case of 110000 (soft tissue face) and 117,500 (skull).

**A. Space Curve Registration To Volume Image Surface:** The trace of  $I_n = \{I_{nx}, I_{ny}, I_{nz}\}$  provides the path along each tile boundary point normal to the voxel representing the segmented surface in the volume image. The position of the detected volume image surface point along the directed normal replaces the  
 5 computed space curve point. Figure 30A shows tiling space curve points registered onto the volume image surface via reference to the linked graphical manifold image. The surface normal at each point is seen as a single perpendicular line.

For each computed tile boundary point in a ridge curve or geodesic B-spline, a nearest point is located on the segmented surface voxels. The surface  
 10 normal vectors, at current curve point  $I$ , are taken as the average of the local normal and interpolated end point normals:

$$I_{nx} = (((I_x - I_{cx}) / xsize) + (e1_{nx} \cdot (1 - p) + e2_{nx} \cdot p)) / 2 \quad \dots \text{(EQ 28)}$$

$$I_{ny} = (((I_y - I_{cy}) / ysize) + (e1_{ny} \cdot (1 - p) + e2_{ny} \cdot p)) / 2 \quad \dots \text{(EQ 29)}$$

$$I_{nz} = (((I_z - I_{cz}) / zsize) + (e1_{nz} \cdot (1 - p) + e2_{nz} \cdot p)) / 2 \quad \dots \text{(EQ 30)}$$

15

where  $xsize$ ,  $ysize$ , and  $zsize$  are the width, height and slices in the volume image.  $I_{cx}, I_{cy}, I_{cz}$  is the midpoint of each coordinate, computed during the loading of a graphical manifold produced from the external tile boundary points. The  $I_{cx}$ ,  $I_{cy}$ , and  $I_{cz}$  are:

$$I_{cx} = \frac{1}{T} \sum_{i=1}^T V_{ix}, \quad I_{cy} = \frac{1}{T} \sum_{i=1}^T V_{iy}, \quad I_{cz} = \frac{1}{T} \sum_{i=1}^T V_{iz} \quad \dots \text{(EQ 31)}$$

20

where  $e1_n$  and  $e2_n$  are the tiling space curve end point (i.e., Type II landmarks) normals, and  $P$  is the distance ratio of the current position  $I$  to the nearest space curve end point.

**B. Interior Tile B-spline Space Curves Extraction:** The internal surface tile  
 25 point vertices are determined via bi-linear Coons surface interpolation (equation 16). The Coons surface is converted into set of interior tile B-spline space curves along  $u$  and  $v$  parametric directions. The normals of each surface tile point are

computed (equations 16, 28, 29, and 30). These normals are used to index the tile points to the volume image surface voxels (Figure 30B). We compute the normal at each parametric surface tile point by directing a trace from the  $u$ ,  $v$  location calculated from the Coon's surface to the voxels representing the segmented surface in the volume image. The detected volume image surface point replaces the Coons surface tile point. Figure 30C shows the extracted surface as it is displayed on the graphical manifold for the operator's approval.

#### **4. Precision and Accuracy Tests and Results**

The SASE 3D surface extraction program was written in C in the Silicon Graphics Unix (IRIX) environment and tested on a SGI Octane workstation (R10000 CPU, 256 Mbyte RAM). User interaction occurs via X11/Motif protocols.

We tested the SASE program with five 3D CT data sets (B2537, B2558, B2621, B3095, B3195) provided by a recall of subjects to the Bolton-Brush Growth Study Center at Case Western Reserve University. All subjects were females above 60 years in age. The boney skull and soft tissue face surfaces were segmented in a separate program. We measured the precision of our ridge curve-based deformable template registration and surface extraction by repetition. We compared both our surface extraction and that obtained by the toolkit to the segmented surface voxels in the volume image to test inter-method accuracy.

**A. Precision Tests:** Intra-operator precision was tested by repeating the SASE template registrations and surface extractions. Direct differences between the two surface extractions are projected into three orthogonal views (Figure 31). Although, differences were minimal, most were localized to the oral and orbital regions. The oral region imprecision may be due to dental artifact. The orbital region imprecision may be due to surface ambiguity in thin bone structures. Operator manual effort in correcting ridge curve extraction varied requiring between 10% (soft tissue face) and 30% (skull) of the total surface extraction time in the SASE method. No tiling curve correction was necessary. We also computed the square root mean of sum of squared distance between all tile points (Table IX).



Figure 33B shows the same color coded difference images of SASE skull and soft tissue face surface extractions. The error is scaled from lower difference in blue to larger differences in red. Note that despite higher sampling density the internal tile points are as true as the tile borders in the SASE extractions. Table XI presents the same results tile by tile. Note that the larger tiles have the greatest error in the toolkit extractions, especially on the calvarium.

- C. Other Inter-method Differences:**
1. Tile Point Density: Figure 34 shows the tile point density. The toolkit extracts fewer points than the SASE program.
  2. Surface Patch Continuity: Continuity of surface curvatures across surface tile boundaries is a critical rendering issue for parameterized ridge curve-based deformable template surface extraction. The toolkit interpolates surface normals providing smooth boundary transitions (i.e., no seam). Figure 35 displays the toolkit and SASE surface extractions. The SASE program avoids interpolation, using the unaltered, detected segmented voxel surface normals.
  3. Tendency of Toolkit to Enlarge Ridge Features: As seen in Figure 32, the toolkit tends to produce a bowing (see chin cleft, reduced nasal aperture, and reduced orbital aperture) of some ridge curves. This is primarily due to the interpolation process for surface tile boundary points beside Type II landmarks.
  4. Time Taken: Skull surface extraction with the toolkit for an experienced operator requires 2-3 hours. In the SASE interface, the operator may extract a skull surface in less than one hour, including the manual effort in correcting 10 to 30% of the automated ridge curve extractions.

### III. Averaging

#### 1. Parametric Surfaces and Surface Extractions

##### A. Surface Parameterization

We use a semi-automatic surface parameterization program, Simulated Annealing Surface Extraction (SASE), to homology map the segmented volume image surface. The SASE surface parameterization occurs via superimposition of a ridge curve-based deformable template to each sample member's surface of interest. The first step in the SASE method is manual location of Type II landmarks on a graphical manifold representation of the segmented surface voxels. Second, the candidate ridge curve-based deformable template is warped to these landmarks as in the toolkit. Finally, ridge and tiling curve superimposition occurs via a simulated annealing algorithm which primarily evaluates surface curvature information found in the unsubsampled volume image data. The superimposed ridge curve-based deformable template defines a homology mapped parametric surface that is encoded as a series of B-spline space curves. The toolkit does not use volume image data to superimpose ridge curve-based deformable templates, relying instead on the operator to manually nudge the ridge curve deformable template onto a graphical manifold.

##### B. B-spline Ridge and Tiling Curve Wireframe

A B-spline space curve may be recorded as a set of 3D  $\{Q_k\}, (k = 1, \dots, n)$  points incrementally stacked along a sweep in 3D space represented as a  $P$ th degree non-rational B-spline. By assigning a parameter value,  $u_k$ , to each  $\{Q_k\}$ , with an appropriate knot vector  $U = \{a, \dots, a, u_{p+1}, \dots, u_{m-p-1}, b, \dots, b\}$ , where  $a$  and  $b$  are initial and final conditions (i.e., 0 ... 1), and  $m$  is an ascending function, a B-spline space curve is:

$$Q_k = C(u_k) = \sum_{i=0}^n N_{i,p}(u_k) P_i \quad \dots \text{(EQ 32)}$$

where  $N_{i,p}(u_k)$  is the  $P$ th degree B-spline basis function. These basis functions define the non-periodic and non-uniform knot vector,  $U$ , and control points,  $P_i$ .

We set the degree of interpolation components as 3, and order 4 (degree + 1) for all space curves. B-spline basis functions of the 3rd degree are defined as:

$$N_{i,0}(u) = \begin{cases} 1, \dots, u_i < u < u_{i+1} \\ 0, \dots, (u \leq u_i) \text{ or } (u \geq u_{i+1}) \end{cases} \dots \text{(EQ 33)}$$

$$N_{i,3}(u) = \frac{u - u_i}{u_{i+3} - u_i} N_{i,2}(u) + \frac{u_{i+4} - u}{u_{i+4} - u_{i+1}} N_{i+1,2}(u) \dots \text{(EQ 34)}$$

5

from the initial condition, these interpolant variables are computed with the knot vectors.

We use a chord length method to select each knot vector and a linear averaging method to determine knot spacing along the ridge curve and tiling curve B-spline space curves. Knot density is pre-determined by the operator. Since each axis is interpolated separately, as in the 2 dimensional case, the x, y, and z axes can be treated separately to encode each space curve.

### C. B-spline Surface Tile Grids

Based on ridge curve and geodesic tile (tiling curve) boundaries alone, a surface tile is defined as a bi-linearly blended Coons surface. We extend this definition. We take the internal tile point grid and tie it to the surface voxels segmented from the original volume. Next, we encode these points as row and column B-spline curves in u, v space (Figure 36A). This improved homology mapping prevents overlapping of internal tile points during averaging. Our surface tiles are either three or four sided. In the case of a three sided tile, the fourth side is generated by placing the same number of second side points, with each point filled by a common degenerate value from the intersection of the first and third boundary space curves (Figure 36B).

Our template uses a bi-linear blending function to define location for interior tile points:

$$Q(u, w) = \begin{bmatrix} 1-u & u & 1 \end{bmatrix} \begin{bmatrix} -P(0,0) & -P(0,1) & P(0,w) \\ -P(1,0) & -P(1,1) & P(1,w) \\ P(u,0) & P(u,1) & 0 \end{bmatrix} \begin{bmatrix} 1-w \\ w \\ 1 \end{bmatrix} \dots \text{(EQ 35)}$$

where the functions  $((1-u), u, (1-w), w)$  (i.e., blending functions) blend the boundary space curves to produce the preliminary internal shape of the surface as a cosine function;  $P$  is the parametric function representing the interior tile points with respect to parametric values  $u$  and  $w$ . The surface points generated are fitted  
 5 to the original image surface via a surface normal trace to the surface voxels segmented from the volume image. The trace modifies the shape of the tile by registering each final surface tile point to the original volume image surface (i.e., segmented voxels). The operator sees the result projected onto a graphical manifold surface. The surface tile points are recorded as rows of B-spline space  
 10 curves spanning the two pairs of tile bounding space curves (Figure 36).

The caption to Figure 37 lists the Type II landmarks found in both soft tissue face and boney skull surface templates. Our current soft tissue face template consists of 56 landmarks, 103 space curves (64 ridge curves and 49 geodesics), and 48 surface tiles (Figure 37A). Our current skull template consists  
 15 of 54 landmarks, 104 space curves (40 ridge curves and 64 geodesics), and 47 surface tiles.

## **2. Computational Tools for Average Generation**

As with the toolkit, the first step in our surface averaging algorithm is the generation of an average Type II landmark configuration. We use the  
 20 Procrustes superimposition method to produce an average Type II landmark configuration. The ridge and tiling space curve B-splines making up each sample member are globally warped to this average landmark shape and locally unwarped.

### **A. Procrustes Superimposition For Production of Average Type II Landmark Configuration**

25 The Procrustes superimposition method is applied to two landmark coordinate configurations at a time. The two homologous shapes are referred to as landmark configurations  $X_1$  and  $X_2$ , are  $p \times k$  matrices with  $k$ -dimensional coordinates of the  $P$  vertices of Type II landmarks. Since the geometric operations are linear, this could be extended to include any number of  
 30 configurations, from  $X_2$  to  $X_N$ . The order in which Procrustes Fitting occurs has



no effect on the shape of the final configuration. A Procrustes fit attempts to minimize the sum of the squared distances between corresponding vertices of both configurations.

Translation: The coordinates of  $X_1$  and  $X_2$  are translated to a shared origin, thereby  $k$  dimensions are reduced from the overall  $pk$  dimensions. This is achieved by subtracting the respective means of each axis from the  $X_1$  and  $X_2$  matrices.  $X_1$  and  $X_2$  are translated to the origin by pre-multiplying  $X$  by  $(I - P)$ , where  $P$  is a  $p \times p$  identity matrix, and is a matrix with every element equal to  $1/p$ . Therefore  $I - P$  is:

$$10 \quad \begin{bmatrix} 1 - \frac{1}{p} & -\frac{1}{p} & -\frac{1}{p} & -\frac{1}{p} \\ -\frac{1}{p} & 1 - \frac{1}{p} & -\frac{1}{p} & -\frac{1}{p} \\ -\frac{1}{p} & -\frac{1}{p} & 1 - \frac{1}{p} & -\frac{1}{p} \\ -\frac{1}{p} & -\frac{1}{p} & -\frac{1}{p} & 1 - \frac{1}{p} \end{bmatrix} \dots (\text{EQ 36})$$

and the translation of  $X_1$  and  $X_2$  to a shared origin is computed as  $X'_1 = (I - P) \cdot X_1$  and  $X'_2 = (I - P) \cdot X_2$ . This translation is shown in Figure 38A.

Scaling: A scalar value, called "Centroid Size,"  $S_1$  of  $X_1$ , is calculated by taking the square root of the sum of squared distances of each landmark to the centroid. Centroid size is computed for each axis separately and divided for each control point  $P$ :

$$S_1 = \sqrt{\text{trace}((I - P)X_1 X_1' (I - P))} \dots (\text{EQ 37})$$

Similarly,  $S_2$  of  $X_2$  is calculated. The translated and scaled

versions  $X_1'' = \frac{1}{S_1} \cdot X'_1$  and  $X_2'' = \frac{1}{S_2} \cdot X'_2$  are seen in Figure 38B.

Rotation: The translated and scaled configuration is next rotated to obtain a minimum root sum of square distance between corresponding points. The first step is to rotate all configurations about the first configuration. Next the "rough"



The consensus average landmark configuration  $X_A$  is computed from  $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_N$  as:

$$X_A = \frac{1}{N} \{ \hat{X}_1 + \hat{X}_2 + \dots + \hat{X}_N \} \quad \dots(\text{EQ 43})$$

## B. Thin Plate Splines

5 We use a thin plate spline to initially position all sample member surfaces at the average Type II landmark configurations. The first landmark transformations matched homologous regions of two and three dimensional wireframe drawings of organisms. A spatial interpolation routine, widely used in geology and metallurgy, the thin plate spline, was suitably modified and applied to  
10 this problem. A thin plate spline warp results in a bending energy matrix describing the shape deformation of the anchor points (i.e., control points and Type II landmarks). We use the thin plate spline to provide a global warp of sample member's ridge, tiling curve, and internal tile B-spline points to the rough average Type II landmark configuration.

15 Although the warp is based only on the Type II landmarks shared by all sample members, the appropriate position of all surface tile points is achieved via the mapping function  $f(R^2 - R^2)$  such that  $f(S_1) = S_A$ , based on the Type II landmarks. This function, a bi-harmonic equation  $U(r) = r^2 \log r^2$ , is a generalized solution. The basis function,  $r = \sqrt{x^2 + y^2}$  is limited to the two  
20 dimensional case. We extend it to three dimensional surfaces as  $U(r) = r$ , where  $r = \sqrt{x^2 + y^2 + z^2}$ . A 3D thin plate spline function can be generalized and interpreted visually as  $\{(r, s, t, \delta(r, s, t)) \in R^4 \mid r, s, t \in R\}$ . If  $S_A$  is our consensus average Type II landmark reference configuration, with  $k = 3$  dimensions and  $p$  vertices, (i.e., a  $k \times p$  matrix), the matrix  $S_A$  and function matrix  $K$  are defined  
25 as:

$$S_A' = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_p \\ y_1 & y_2 & y_3 & \dots & y_p \\ z_1 & z_2 & z_3 & \dots & z_p \end{bmatrix}, 4 \times p \dots \text{(EQ 44)}$$

$$K = \begin{bmatrix} 0 & U(r_{12}) & \dots & u(r_{1p}) \\ U(r_{21}) & 0 & \dots & U(r_{2p}) \\ \dots & \dots & \dots & \dots \\ U(r_{p1}) & U(r_{p2}) & \dots & 0 \end{bmatrix}, p \times p \dots \text{(EQ 45)}$$

$U(r_{ij})$  is defined in 3D as the distance from the  $i^{th}$  to  $j^{th}$  the point in the reference configuration as:

$$U(r_{ij}) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \dots \text{(EQ 46)}$$

Next, a projection operator  $L$  is defined as:

$$L = \begin{bmatrix} K & S_A \\ S_A' & O \end{bmatrix}, (p+4) \times (p+4) \dots \text{(EQ 47)}$$

$O$  is a 4x4 null matrix. We use these mapping functions to globally warp a series of ridge curve based deformable template (i.e., ridge, tiling, and internal tile B-spline space curves) wireframes to the rough average Type II landmark configuration via a global warp. Once in position, each space curve is locally unwarped to begin the averaging process.

In general, thin plate spline warping is defined with a set of homologous points  $Y$ , as  $L^{-1}Y$ , and the transformation is defined:

$$S(X'_m) = A \begin{bmatrix} 1 & x & y & z \end{bmatrix}'_m + \sum_j^p n_j U(r_j), n=1 \dots N, i=1 \dots I \dots \text{(EQ 48)}$$

### 3. Average Surface Generation

Our space curve averaging is a two step process. The first step begins following the local unwarping of space curves that were globally thin plate spline warped to an average landmark configuration. Then all sample members are locally unwarped and averaged, B-spline sample by B-spline sample (i.e., each set of homologous ridge, tiling, or internal tile curves).

### A. Average Type II Landmark Configuration

We use the Procrustes Superimposition method to align the sample member landmark configurations and compute an average configuration from them. Procrustes fitting removes variation due to position, orientation, and scale prior to our calculating a simple average. In equation 45 we see that the computed  
 5 average configuration is set to  $X_A$ .

### B. Average Space Curve Generation

This is accomplished algorithmically as follows: Let the configurations  $S_n$  where  $n$  varies from 1 to  $N$  wireframe space curves (i.e., The  
 10 wireframe consists of the tile boundary B-spline space curves).  $X_A$  is an average landmark configuration (Equation 45). Each sample configuration  $S_n$  is warped to  $X_A$  landmarks. We refer to this sample of globally warped configurations as  $\hat{S}_n$ .  
 Next, each member of  $\hat{S}_n$  is locally unwarped to the original configuration  $S_n$  for each space curve. The locally unwarped and assembled space curves are shown in  
 15 Figure 39 drawn between hypothetical Type II landmarks A and B. A "rough" average curve is produced as a direct average of the control point coordinates (note: this is possible for B-splines but not for Bezier splines where the control points may not lie on the encoded space curve). The rough average curve is then re-splined. The re-splined rough average curve is displayed as a dashed line in  
 20 Figure 39. The second step is the generation of final average space curves. Sampling planes (perpendicular bi-sectors) are positioned at regularly spaced intervals along the rough average curve. A control point for each sampling plane is produced from the average of intercepted points. Finally, a B-spline space curve is generated from the average intercept points (Figure 39, thick dark line).

25 The space curve averaging is done separately for each type of space curve (i.e., ridge, tiling, and internal tile curves). Ridge curves are further divided into arc-segments between Type II landmarks for averaging. The complete set of average space curves represents the average surface.

### C. Internal Tile Curve Averaging

Internal tile curves have either  $u$  or  $v$  orientation (see Figure 36). Therefore each internal tile point belongs to two B-splines. We compute the average  $u$  and  $v$  internal tile B-spline space curves separately. The average of these two points is then substituted back to internal tile vertices, in order to prepare a triangulated (graphical manifold) surface. Algorithmically, B-spline internal tile curve segments in the  $u$  and  $v$  direction may be represented as matrices:

$$S_t(C_u) = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,v} \\ C_{2,1} & C_{2,2} & \dots & C_{2,v} \\ \dots & \dots & \dots & \dots \\ C_{u,1} & C_{u,2} & \dots & C_{u,v} \end{bmatrix}, \quad S_t(C_v) = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,u} \\ C_{2,1} & C_{2,2} & \dots & C_{2,u} \\ \dots & \dots & \dots & \dots \\ C_{v,1} & C_{v,2} & \dots & C_{v,u} \end{bmatrix} \dots \text{(EQ 49)}$$

The computed interior surface tile control points thus are the average of the above two matrices,  $S_t(C_u)$  and  $S_t(C_v)$ :

$$S_t(C) = S_t(C_u) + S_t(C_v)' = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,v} \\ C_{2,1} & C_{2,2} & \dots & C_{2,v} \\ \dots & \dots & \dots & \dots \\ C_{u,1} & C_{u,2} & \dots & C_{u,v} \end{bmatrix} + \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,u} \\ C_{2,1} & C_{2,2} & \dots & C_{2,u} \\ \dots & \dots & \dots & \dots \\ C_{v,1} & C_{v,2} & \dots & C_{v,u} \end{bmatrix} \dots \text{(EQ 50)}$$

where  $t$  varies from 1 to the number of tiles in the whole sample surface.

### 4. Precision and Inter-method Tests and Results

The SSA program was written in C in the Silicon Graphics Unix (IRIX) environment. It was tested on a SGI Octane workstation (R10000 CPU, 256M byte RAM). The program interface integrates Procrustes superimposition, thin plate spline warp, and surface averaging methods with graphical manifold surface rendering operator feed back. User interaction occurs via X11/Motif protocols.

To test the SSA program, we averaged soft tissue face surfaces (Figure 40A) and skull surfaces (Figure 40B) from segmented from 3D CT volume images. These five 3D CT data sets (B2537, B2621, B3037, B3095, B3195) were provided by a recall of subjects. All subjects are above 60 years in age, female,

have dental fillings, and often have large oral prostheses (dentures). The soft tissue face and skull surface segmentation of the volume images was performed in a separate program, SOFM. The ridge curve-based deformable template superimposition and surface parameterization was done in an another program, SASE.

**A. Intra-Operator Precision Test**

We conducted a precision test of the SSA methods between two sessions by one operator. The operator introduces variation through initial seeding and subsequent fine tuning of the ridge and tiling curve template superimposition. The overall root mean square error between the surface tile points of two averaging sessions, of the boney skull surface average was 0.0 mm and for the soft tissue face surface average was 0.0 mm. This is complete agreement between sessions. This intra-operator reproducibility result (i.e., precision) provides evidence that the SSA method is reliable.

**B. Inter-method Tests**

We next compared the surface sampling used to produce average surfaces by the SSA program versus that of the toolkit using the same segmented voxel data obtained from the SOFM program. We hypothesize that improved homology of tiling and internal tile curves in the SSA method will reduce the variance of the sample members about the average.

**i. Visual Comparison**

We parameterized all 10 surfaces with both the SASE and the toolkit for the soft tissue face (Figure 41A) and the boney skull (Figure 41B). Note that the toolkit sample and average surface images appear to bow along every ridge curve edge (e.g., orbits and nasal regions). Under the chin, we note that this problem has exaggerated a notch not seen in the SSA average. This bulging also results in smaller orbital and nasal apertures relative to the shape seen in the original segmented surface voxels. This and the smoother surface may be an artifact of splining a lesser number of points than in the SSA method.

**ii. Inter-method Comparison of Average Surface Warped to Members' Original Segmented Voxel Surface**

We thin plate spline warped the average surface to the Type II landmark configuration of its member surfaces. Then, we traced each fitted average surface tile point to the segmented surface image. Both averaging methods (i.e., toolkit and SSA) used surfaces extracted from the same segmented volume image, providing a common basis for comparison between the two methods. The computed square root of sum of squared differences are listed in Table XII. Note, the number of tile points in the toolkit surface extractions is well below that used in the SSA method. Given this difference, we observed an average of 6 mm distance for the toolkit averaging method for all points in the soft tissue face and skull surfaces, whereas the average SSA distance were 0.448 mm in the case of the soft tissue average and 0.419 mm in the case of the skull average. Table XIII presents these results tile by tile. Figure 42A displays color coded difference images of toolkit skull and soft tissue face surface averages with their source segmented voxel surfaces. Similarly Figure 42B shows the same color coded difference images of the SSA skull and soft tissue face surface averages. The same range of difference is scaled from lowest difference in blue to largest difference in red in both images. Note that despite higher sampling density the internal tile points are as true as the tile borders in the SSA extractions.

### iii. Results

· In Figure 43A, we see the toolkit average surface prior to the final step of normal interpolation and averaging. Figure 43B presents the same surfaces afterward. Figure 43C presents the final SSA average surface; the SSA algorithm  
25 does not average surface normals. Note that seams are present but they less disturb the continuity of surface curvature between the tile surfaces. In both cases these are graphical rendering artifacts, not an indication of inaccuracy in surface parameterization or a non-shape preserving average.

## 5. Conclusion

30           One reason that the SSA average surface results better preserve  
surface curvature continuity, and reduce variance about the average, is the higher



sampling density (Figure 44). The toolkit's graphical manifold is subsampled at four stages. Surface rendering of the segmented voxels introduces an error of  $\pm 1$  to  $\pm 3$  mm assuming a sub-sampling rate of  $2 \times 2 \times 2$ .

5       Second, a wireframe is manually nudged into best position on this graphical manifold and then interpolated. This step increases the error at each point by  $\pm 1$  to  $\pm 3$  mm depending on the variable level of parametric smoothing that the program automatically applies. Third, the surface is sparsely encoded. This may eliminate some otherwise useful information from the patient surface image.

10       Finally when generating the average, only the surface tile boundaries (i.e., ridge curve and geodesic lines) are truly averaged in the toolkit. The average interior tile points are assigned, not found. Overall, the errors accumulated at each surface tile point vary from 3 to 6 mm.

15       It appears to us that the SSA method improves tiling and internal tile curve homology assignment. The SSA method better preserves overall shape of 3D surfaces during averaging because of (1) the improved homology assignment of internal tile points, and (2) the extension of the original space curve averaging method to the entire surface.

20       The utility of average surface images for clinical diagnosis needs validation. Their use for boney prosthetic design is apparent. These methods were originally developed to quantify diagnostic morphometric differences of clinical populations, however we expect these data have applications at every stage of patient care, as well as other deformable model applications including animation.

25       The invention has been described with reference to the preferred embodiment. Obviously, modifications and alterations will occur to others upon reading and understanding the preceding detailed description. It is intended that the invention be construed as including all such modifications and alterations insofar as they come within the scope of the appended claims or the equivalents thereof.